



Hello Alessandro,

If you are reading this we clearly think you can be an awesome part of our team 😎.

We strongly believe in providing to developers the best environment and the best tech tools for them to do what they like, developing great software!

To accomplish this, we would give you a code challenge that will help us to better understand your skills to find out whether or not we are a good fit for each other.

The Code Challenge

Giulio, on a random normal day, bursts into our “Dev Room” and starts to explain to us his last theory about “office happiness”.

He thinks that a system of employee reviews can increase team harmony.

Functional requirements

Write a web application (based on Laravel) that lets employees write anonymous reviews of other employees. Given an employee email (you can assume that an email of the type username@corsi.it is given as an identifier of a specific user), show a web page with a classic 5 stars and textarea review form where the viewing user can submit an employee “happiness review”. The form should also show a dropdown menu where the choice of the employee to review can be made (you can generate sample employees however you want for the purpose of the challenge).

So mainly, you’ll have to implement three views: the login view (with just the email), the review view (5-stars, select and textarea), and the final thank you view. For the backend part please expose a REST api that gives back the full list of the reviews stored in the system.

Too easy? Show off your skills

If you wanna give a shot to the best skills of yours, you may think to build an extra view (after the email based login view) that shows a table containing a list of reviews filtered by email & rating, sorted by email or rating, and with pagination.

Then, you’ll add the possibility to create a review from the command line with something like `php artisan reviews:create --email=xxxx --review=xxxx --rating=xx`.

Lastly, since we care about respect in the workplace, you're going to make sure that creating a review containing the following unkind words fails:

Boring, ugly, smelly, shitty, .NET, bullshit, he owes me money

The top player zone - frontend

If you are here, and you want definitely smash our expectations, implement a view that shows a ranked list (based on the average rating) of employees and a line graph showing the daily "happiness index", which is the average rating among all reviews in a given day, showing at least the last 15 days (Be sure to provide us with a fake data generator to see this in action!). Even cooler, this page should update automagically when new data arrives.

The top player zone - backend

Ok, now for a more architectural challenge: your app has to integrate with TotallyRealCompanyReviews.com, the not-at-all-just-invented company reviews public database. Two things are going to happen:

- TRCR is going to send you reviews about your company for you to store. It's going to try to reach the endpoint [yourapp/api/webhooks/company-reviews](#) with the following JSON body request format:

```
{"company": "Corsi.it", "email": "xxxx", "review": "xxxxx", "rating": x, "date": "xxxx-xx-xxTxx:xx:xxZxxx"}
```

- Your app will have to send each new review to TRCR's database by making a POST request to <https://totallyrealcompanyreviews.com/api/reviews> with the following JSON body request format:

```
{"company": "Corsi.it", "email": "xxxx", "review": "xxxxx", "rating": x, "date": "xxxx-xx-xxTxx:xx:xxZxxx"}
```

N.B. Don't actually implement TRCR's API, just pretend it exists

Non functional requirements

You will be working on a big and quite complex system and we would like to share the same mindset among developers in order to avoid incongruences and limit bugs (everyone hates bugs 🐛).

For these reasons we would like you to implement the code challenge taking into account that:

- we work Agile with PHP8, Vue/React
- we love RESTful API that every client can implement with no problems (we hate non standard implementations)
- we make full usage of GitHub
 - please commit in the way you're used to

- we love clear instructions accompanying great software, especially on how to run the app, run the tests, update projects and libraries
 - we really love README.md ❤️
- we hate static libraries, we love package managers
 - we don't like to include stuff, we compose and import software

This code challenge should take 2-4 days to be completed 💪, this is generally the time other applicants needed to complete it. Please choose the right trade off between speed and quality and take note of the rationale behind your decisions, it might be useful later. *Don't be afraid to ask questions even during the development if some requirements turn out to be fuzzier than expected, just as you would do at work.*

Start by cloning this [repo](#) and when you're done, PR your work!

We will check the web app you developed and one of our devs will reach out to you to review your work together 👍.

Looking forward to pulling your awesome code and good luck 🍀.

Thanks,

Corsi.it Dev Team